Machine Learning Theory (CSC 482A/581B) - Lecture 16

Nishant Mehta

1 Occam's razor bounds

Thus far in this course, we have obtained risk bounds via a uniform convergence approach: we have proved that with high probability, uniformly over all hypotheses in some set, the actual risk is not much greater than the empirical risk. Specifically, we showed guarantees of the form:

With probability at least $1-\delta$, for all $f \in \mathcal{F}$:

$$\Pr_{X\sim P}\left(f(X) \neq Y\right) \leq \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}\left[f(X_j) \neq Y_j\right] + \varepsilon.$$

In this type of bound, ε depends on n, δ, and $\mathcal{F}$; however, regardless of the hypothesis f being considered, the same quantity ε always appears in the above bound. It is in this sense that we refer to this as a uniform bound.

How did we obtain bounds of this form? In the case of a finite set of hypotheses $\mathcal{F}$, we essentially just applied Hoeffding's inequality (in the agnostic case) or a Binomial tail bound (in the realizable case) to show concentration of the empirical risk around the actual risk of a single hypothesis, and then we used the union bound to ensure such concentration holds for all hypotheses simultaneously.

In this lecture and the next, we will focus on non-uniform bounds. The first type of bound we will explore takes the form:

With probability at least $1-\delta$, for all $f \in \mathcal{F}$:

$$\Pr_{X\sim P}\left(f(X) \neq Y\right) \leq \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}\left[f(X_j) \neq Y_j\right] + \varepsilon(f).$$

In the above bound, ε has now been replaced with a quantity $\varepsilon(f)$ that can vary with each individual hypothesis. As we will see, the reason for letting ε vary with f is that we can obtain a better bound than the uniform case for some f. This improvement can be useful to a practitioner that has prior beliefs over which hypotheses are more likely to be output by a learning algorithm However, this improvement is not for free: the price paid will be that the bound necessarily worsens for other f. This trade-off derives from the fundamental connection between prefix codes and probability distributions which we will explore below.

We now develop a nonuniform bound called an Occam's razor bound (or Occam bound) for short. For any hypothesis f, denote by $R(f)$ the risk of f under zero-one loss; that is,

$$R(f) = \Pr_{(X,Y)\sim P}\left(f(X) \neq Y\right).$$

Also, denote by $\hat{R}_n(f)$ the empirical risk of f under zero-one loss, so that

$$\hat{R}_n(f) = \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}\left[f(X_j) \neq Y_j\right].$$

Theorem 1. *Let Π be a probability distribution over $\mathcal{F}$. If $(X_1, Y_1), \ldots, (X_n, Y_n)$ are independently distributed according to probability distribution P, then with probability at least $1-\delta$, for all $f \in \mathcal{F}$,*

$$R(f) \leq \hat{R}_n(f) + \sqrt{\frac{2R(f)\left(\log \frac{1}{\Pi(f)} + \log \frac{1}{\delta}\right)}{n}}.$$

Proof. First, consider a fixed $f \in \mathcal{F}$. We will use the relative Chernoff bound below to upper bound $R(f)$ by the empirical risk of f:

> Let $Z_1, \ldots, Z_n$ be independent Bernoulli random variables and define $S_n = \sum_{j=1}^{n} Z_j$. Then for $\varepsilon \in (0, 1]$,
>
> $$\Pr\left(S_n \leq (1-\varepsilon)\,\mathsf{E}[S_n]\right) \leq e^{-\varepsilon^2\,\mathsf{E}[S_n]/2}.$$

To apply this result, we take $Z_j = \mathbf{1}\left[f(X_j) \neq Y_j\right]$ for $j \in [n]$ and scale the expression inside the probability by $\frac{1}{n}$, yielding

$$\Pr\left(\hat{R}_n(f) \leq R(f) - \varepsilon R(f)\right) \leq e^{-n\varepsilon^2 R(f)/2}.$$

Setting the RHS to δ_f and solving for ε yields

$$\varepsilon = \sqrt{\frac{2\log \frac{1}{\delta_f}}{nR(f)}}.$$

Plugging in ε and rearranging, we have the following high probability guarantee for hypothesis f:

$$\Pr\left(R(f) > \hat{R}_n(f) + \sqrt{\frac{2R(f)\log \frac{1}{\delta_f}}{n}}\right) \leq \delta_f. \tag{1}$$

So far, the above has not really used any new ideas aside from resorting to a relative Chernoff bound. The key novelty of the Occam bound is what we introduce next; the following procedure can be thought of as a weighted union bound. For each $f \in \mathcal{F}$, apply the bound (1) with $\delta_f = \Pi(f) \cdot \delta$. Then the result follows by observing that

$$\begin{aligned}
&\Pr\left(\exists f \in \mathcal{F} : R(f) > \hat{R}_n(f) + \sqrt{\frac{2R(f)\left(\log \frac{1}{\Pi(f)} + \log \frac{1}{\delta}\right)}{n}}\right) \\
&\leq \sum_{f \in \mathcal{F}} \Pr\left(R(f) > \hat{R}_n(f) + \sqrt{\frac{2R(f)\left(\log \frac{1}{\Pi(f)} + \log \frac{1}{\delta}\right)}{n}}\right) \\
&\leq \sum_{f \in \mathcal{F}} \Pi(f)\delta \\
&= \delta.
\end{aligned}$$

□

In applying the above bound, it is critical that the distribution Π is chosen before seeing the training sample. In this sense, we can think of Π as a prior distribution. Note, however, that Π need not be viewed as a Bayesian prior; unlike in Bayesian statistics, Theorem 1 is correct for *any* choice of Π, as long as Π is independent of the training sample.

In the past, we have used the special case of a uniform prior distribution $\Pi(f) = \frac{1}{|\mathcal{F}|}$ and the coarse upper bound $R(f) \leq 1$, which recovers the familiar bound:

With probability at least $1 - \delta$, for all $f \in \mathcal{F}$,

$$R(f) \leq \hat{R}_n(f) + \sqrt{\frac{2\left(\log|\mathcal{F}| + \log\frac{1}{\delta}\right)}{n}}.$$

The bound in Theorem 1 is more powerful though. Suppose that a practitioner believes that some hypotheses are far more likely than others to be returned by their learning algorithm. They can reflect this prior knowledge by selecting a (data-independent!) prior distribution that places more weight on such hypotheses. Then, if they are lucky, the algorithm will indeed return a hypothesis for which $\Pi(f)$ is large and hence for which $\log\frac{1}{\Pi(f)}$ is small. In this lucky event, the practitioner can apply the bound in Theorem 1 for only this f, and the complexity price paid in the bound will be only $\log\frac{1}{\Pi(f)}$ rather than the larger price of $\log|\mathcal{F}|$ (which they would have paid had they selected a uniform prior). As we will soon see, the "price" $\log\frac{1}{\Pi(f)}$ can be naturally interpreted as the length (in bits) used to encode hypothesis f using a code that is based on the probability distribution Π.

2 Codes, probability distributions, and prediction under log loss

As already hinted when discussing the Occam bound, there is a fundamental connection between prefix codes and probability distributions.

Let $\mathcal{X}$ be a finite[1] sample space and let $\mathbb{B}^*$ be the set of all binary strings of any length. A prefix code for $\mathcal{X}$ is a mapping $C\colon \mathcal{X} \to \mathbb{B}^*$ which has the property that, for all outcomes $x, y \in \mathcal{X}$, the code word $C(x)$ is not a prefix of the code word $C(y)$. Why should we restrict to prefix codes? These codes turn out to be the only type of codes for which messages formed by code words can be unambiguously decoded.

Given a code C and an outcome $x \in \mathcal{X}$, we denote the length (in bits) of the code word $C(x)$ by $L_C(x)$.

We say that a probability distribution P over $\mathcal{X}$ is *defective* if $\sum_{x\in\mathcal{X}} P(x) < 1$. Any defective distribution can be made into a proper (i.e. non-defective) probability distribution by augmenting the sample space with an additional dummy symbol $\square$ and setting $P(\square)$ such that $\sum_{x\in\mathcal{X}\cup\{\square\}} P(x) = 1$.

We say that a prefix code C is *complete* if there is no code C' such that both of the following properties hold:

- for all $x \in \mathcal{X}$, $L_{C'}(x) \leq L_C(x)$;
- there exists an $x \in \mathcal{X}$ for which $L_{C'}(x) < L_C(x)$.

Given a prefix code C for $\mathcal{X}$, there exists a (possibly defective) probability distribution P over $\mathcal{X}$ for which $-\log P(x) = L_c(X)$. There is a precise connection between probability distributions

[1] $\mathcal{X}$ is taken to be finite purely for simplicity

and complete prefix codes. A prefix code is complete if and only if its corresponding probability distribution P is not defective. We will not prove this fact here.

It turns out that if we slightly generalize the notion of codes and codelength, there is a converse as well. Suppose that we allow for codes that can have non-integer codelengths; technically, such codes do not exist, so this is an idealization. However, if we speak only in terms of code *lengths*, without actually ever needing to implement the code, this will not cause a problem. From now on, we allow these more general codes, and we will only ever use the codelength function L_C rather than directly using the code C.

Using these idealized codes, we have the desired converse: given a probability distribution P over $\mathcal{X}$, there exists a prefix code C for $\mathcal{X}$ such that, for all $x \in \mathcal{X}$, $L_C(x) = -\log P(x)$.

Thus, using these idealized codes, there is a fundamental correspondence between the codelength functions of (complete) prefix codes and probability distributions:

- given any codelength function L_C for codes over $\mathcal{X}$, there is a probability distribution P over $\mathcal{X}$ defined as $P(x) = 2^{-L_C(x)}$;
- given any probability distribution P over $\mathcal{X}$, there is a codelength function $L_C(X)$ defined as $L_C(X) = -\log P(x)$.

By allowing for idealized codes, we can also extend the notion of a prefix code being complete. The Kraft inequality states that for any (usual / non-idealized) prefix code C over $\mathcal{X}$,

$$\sum_{x \in \mathcal{X}} 2^{-L_C(x)} \leq 1. \tag{2}$$

We will say that an idealized code is complete when $\sum_{x \in \mathcal{X}} 2^{-L_C(x)} = 1$, so that there is no room for improvement.

Consider a game that unrolls according to the following protocol:

1. Alice selects a probability distribution P over $\mathcal{X}$.
2. Bob selects a code C over $\mathcal{X}$ that he will use to encodes samples from P.
3. Alice then draws a sample $X \sim P$ and Bob encodes X, paying a price of $L_C(X)$ bits.

In the above game, Bob's goal is to minimize the expected number of bits he uses to encode samples from P. That is, Bob seeks to minimize

$$\mathsf{E}_{X \sim P} \left[L_C(X) \right].$$

For what code C is the above objective minimized? To answer this question, we will leverage the bijection between codelength functions and probability distributions. Bob's selection of some code C is equivalent to his selection of a probability distribution Q defined via $-\log Q(x) = L_C(x)$. Bob's goal is then to select a probability distribution Q that minimizes

$$\mathsf{E}_{X \sim P} \left[-\log Q(X) \right].$$

Expressed in this way, the problem of selecting a code that minimizes expected codelength is equivalent to the problem of selecting a probability distribution that minimizes expected log loss. There is thus a perfect correspondence to the problem of density estimation under log loss.

The optimal choice of Q turns out to be the distribution P that generated the data. That is, we have the following lemma:

Lemma 1. *For any probability distributions P and Q over $\mathcal{X}$,*

$$\mathsf{E}_{X\sim P}\left[-\log Q(X)\right] - \mathsf{E}_{X\sim P}\left[-\log P(X)\right] \geq 0, \tag{3}$$

and the inequality is strict whenever $P \neq Q$.

Proof. First, we rewrite the LHS as

$$\mathsf{E}_{X\sim P}\left[-\log Q(X)\right] - \mathsf{E}_{X\sim P}\left[-\log P(X)\right] = \mathsf{E}_{X\sim P}\left[-\log \frac{Q(X)}{P(X)}\right].$$

Next, observe that $x \mapsto -\log x$ is a strictly convex function. We will use the following property of strictly convex functions. For any strictly convex function $f\colon \mathcal{X} \to \mathbb{R}$ and any distribution P over $\mathcal{X}$ whose support contains at least at least two elements,

$$\mathsf{E}_{X\sim P}[f(X)] > f(\mathsf{E}[X]).$$

Using the above inequality with $f\colon x \mapsto -\log x$, we have

$$\mathsf{E}_{X\sim P}\left[-\log \frac{Q(X)}{P(X)}\right] \geq -\log \mathsf{E}_{X\sim P}\left[\frac{Q(X)}{P(X)}\right] = -\log \sum_{x\in\mathcal{X}} Q(x) = 0,$$

and the inequality is strict whenever $P \neq Q$. □

Thus, the (idealized) code that minimizes expected codelength has a codelength function for which $L_C(x) = -\log P(x)$.

Using the connection between minimizing expected codelength and density estimation under log loss, we can interpret standard information-theoretic quantities like the entropy and relative entropy.

When using distribution P to code outcomes drawn from P, the expected codelength is precisely equal to the entropy of P.

Definition 1. The *entropy* of a probability distribution P over $\mathcal{X}$ is defined as

$$H(P) = \mathsf{E}_{X\sim P}[-\log P(X)] = -\sum_{x\in\mathcal{X}} P(x)\log P(X).$$

The notion of entropy can be generalized by allowing for possibly suboptimal codes based on some distribution Q not necessarily equal to P. This gives rise to cross-entropy.

Definition 2. The *cross-entropy* between probability distributions P and Q over $\mathcal{X}$ is defined as

$$H(P,Q) = \mathsf{E}_{X\sim P}[-\log Q(X)] = -\sum_{x\in\mathcal{X}} P(x)\log Q(X).$$

The cross-entropy between P and Q is thus the expected number of bits required to encode outcomes from P using a code based on Q.

Finally, the difference (see the LHS of (3)) between the entropy of P and the cross entropy between P and Q is known as the *relative entropy* of P relative to Q, also called the *Kullback-Leibler (KL) divergence* of P from Q.

Definition 3. For probability distributions P and Q over $\mathcal{X}$, the *KL divergence* of P from Q is defined as

$$\mathsf{E}_{X\sim P}\left[-\log Q(X)\right] - \mathsf{E}_{X\sim P}\left[-\log P(X)\right].$$

Interpreted from the lens of density estimation under log loss, KL divergence thus expresses the excess risk due to predicting via distribution Q rather than the optimal distribution P. Also, as shown in Lemma 1, the KL divergence is always nonnnegative and, for fixed P, it is uniquely minimized by taking $Q = P$.